

Odin Detector DAQ Framework and areaDetector Integration

**EPICS areaDetector Working Group Meeting
APS, June 2018**

**Ulrik Pedersen
Head of Beamline Controls**



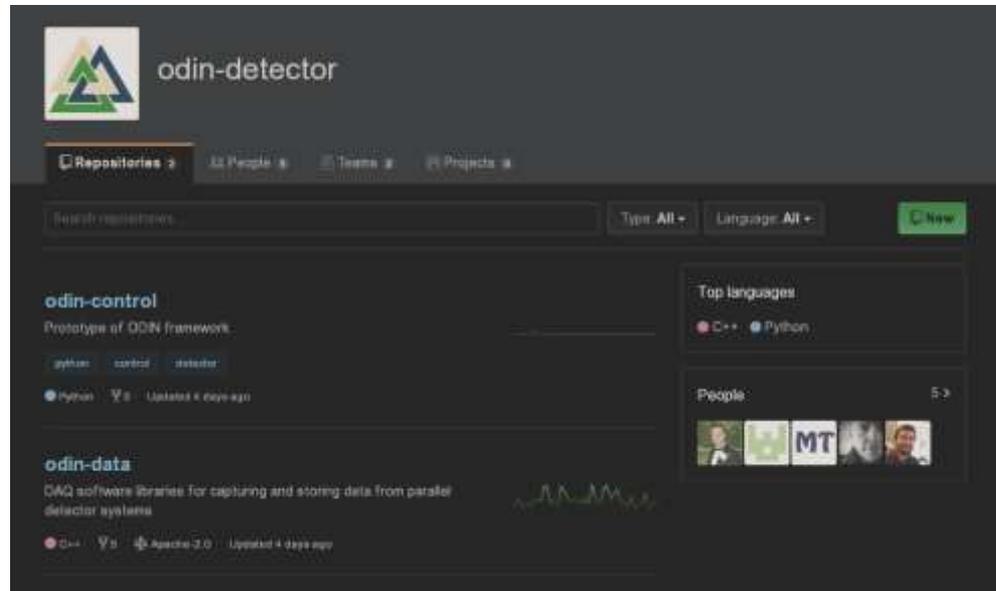
Odin

*“Control and data acquisition framework
for parallel detector systems”*

DLS – STFC Collaboration

Started: Late 2014

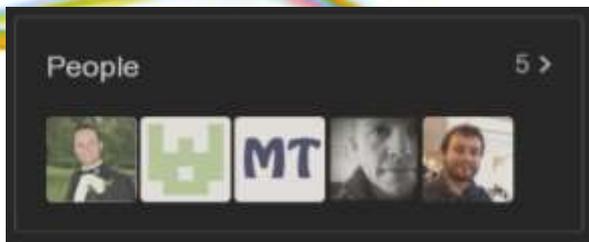
GitHub



Collaborators

Log

Cryptic Name



odin-detector
 diamond

Odin Detectors

- **In-house or collaborative detector development**
 - Different hats: detector system vendor & system integrators
- **Collaboration agreement: control system agnostic**
- **Scalable, but simple to control**
 - Modular detector systems, multiple readout channels
 - Single point of control of distributed system
- **DAQ system, not a processing pipeline**
 - But ... descrambling pixels **is** processing

Odin Detectors



Excalibur



Percival

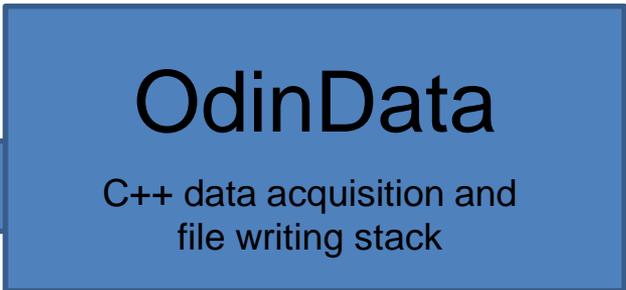
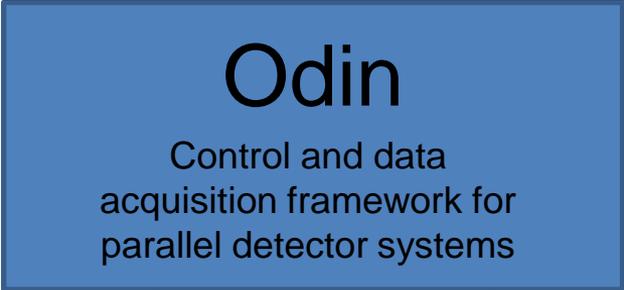


Tristan

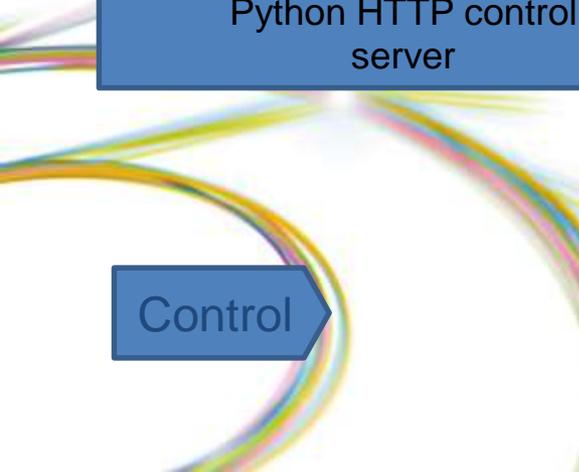


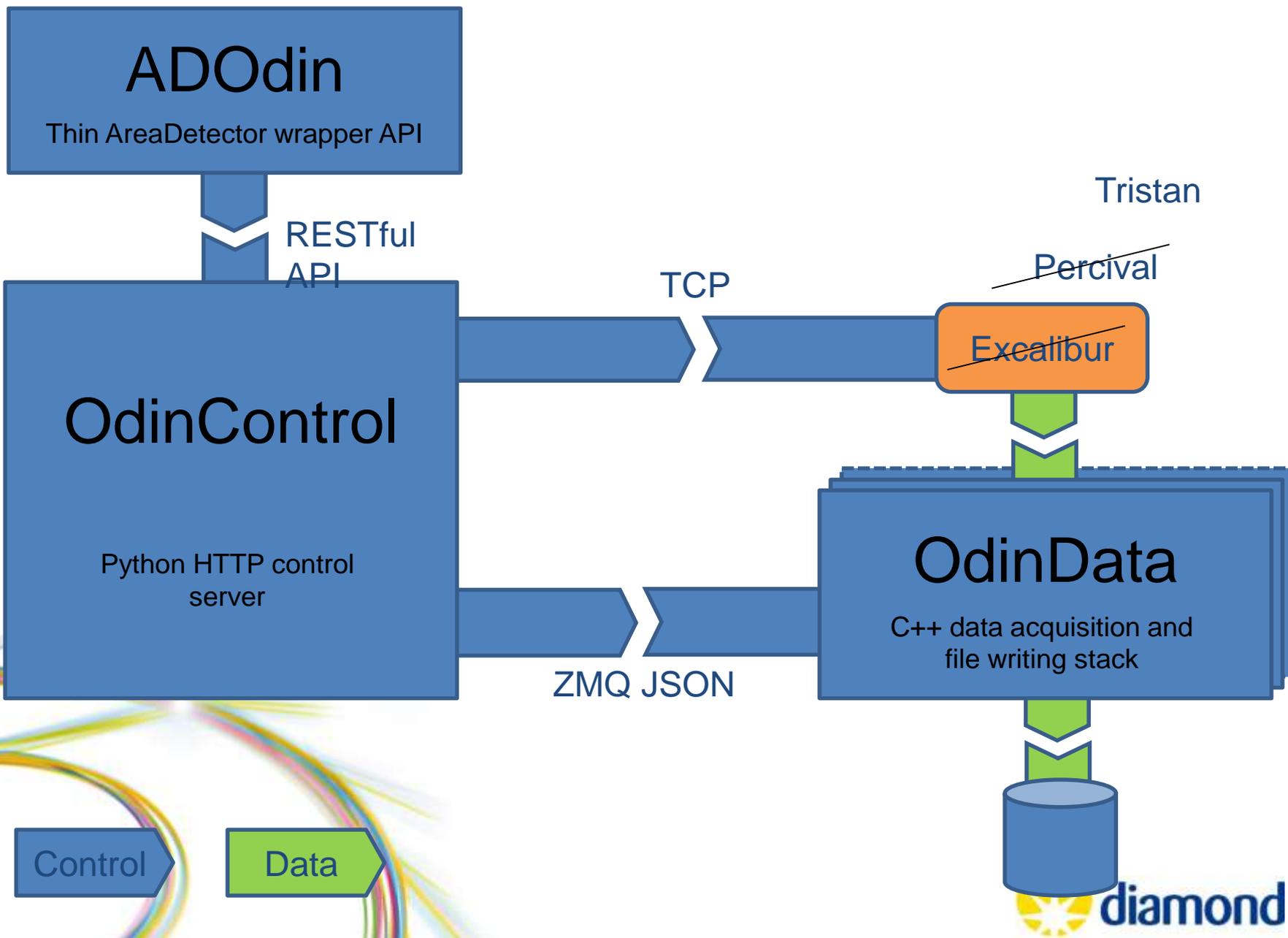
Eiger...

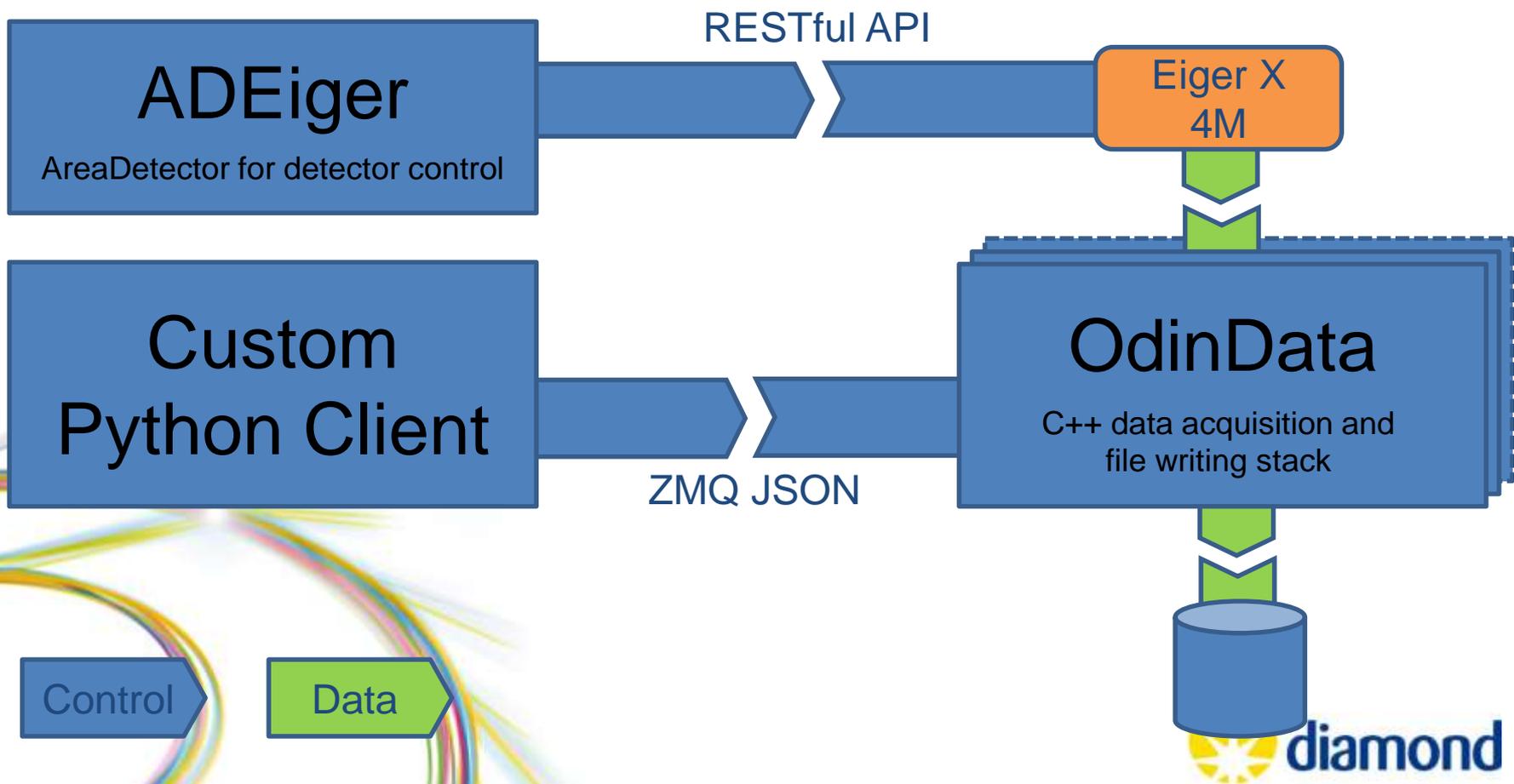




ZMQ JSON







OdinControl

- **Generic HTTP server**
- **Adapters provide functionality for each device**
- **Server generates RESTful API from arbitrary Adapter attributes and methods**
- **Control through simple webpages or client app**

The image shows two screenshots of the OdinControl web interface. The top screenshot displays the 'Server Status' page, which provides a comprehensive overview of the system's operational state. The bottom screenshot shows the 'Control' page, which allows users to interact with the system through various command and configuration options.

Server Status

API Version:	0.1	
Adapters Loaded:	perchval	
Server Start Time:	September 26, 2017 11:54:52	
Server Up Time:	0:07:56.723233	
Server Username:	gnc91527	
Hardware Connection:	Address:	127.0.0.1
	Port:	10001
	Connected:	●
Database Connection:	Address:	127.0.0.1
	Port:	8086
	Name:	perchval
	Connected:	●
Auto-read Monitors (10 Hz):	● <input type="button" value="Stop"/>	

Control

Control Message Response:

```
Command: cmd_system_command
Parameters: name = start_acquisition
Execution Start Time: 2017-09-26 11:58:15.287993
Response: Completed
Message:
```

Download Channel Settings:

Initiate Channel:

System Command:

Set Channel Value:

Apply Set Point:

Upload Configuration File:

OdinControl Server Configuration

- **Configure server address**
- **Load Adapter libraries**
- **Configure specific Adapters**

```
[server]
debug mode = 1
http_port = 8888
http_addr = 0.0.0.0
adapters = excalibur, odin_data

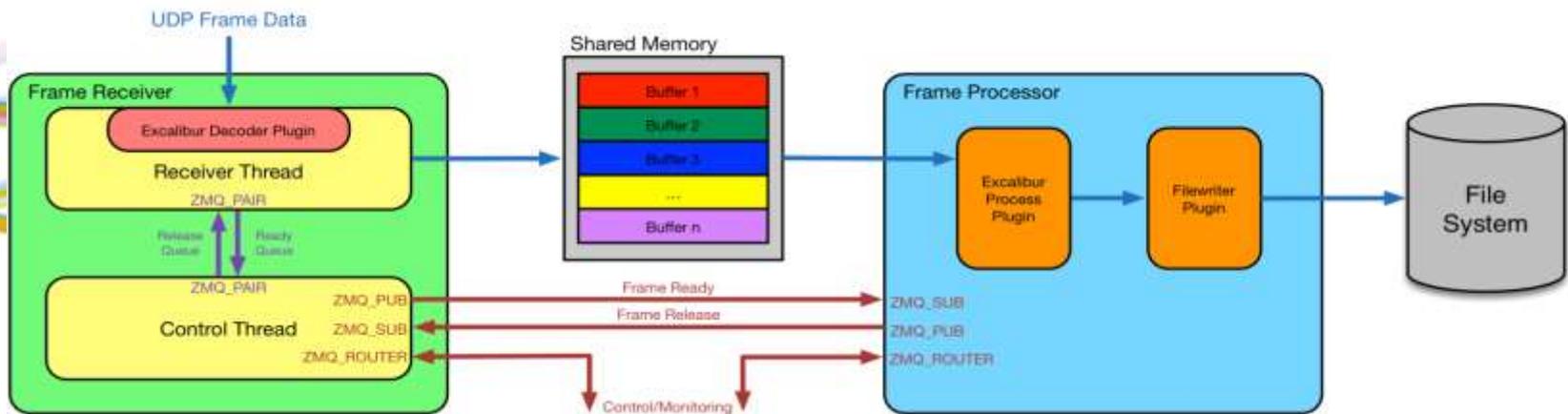
[tornado]
logging = debug

[adapter.excalibur]
module = excalibur.adapter.ExcaliburAdapter
detector_fems = 192.168.0.1:6969, 192.168.0.2:6969

[adapter.odin_data]
module = odin_data.odin_data_adapter.OdinDataAdapter
endpoints = 127.0.0.1:5004, 127.0.0.1:6004
update_interval = 0.5
```

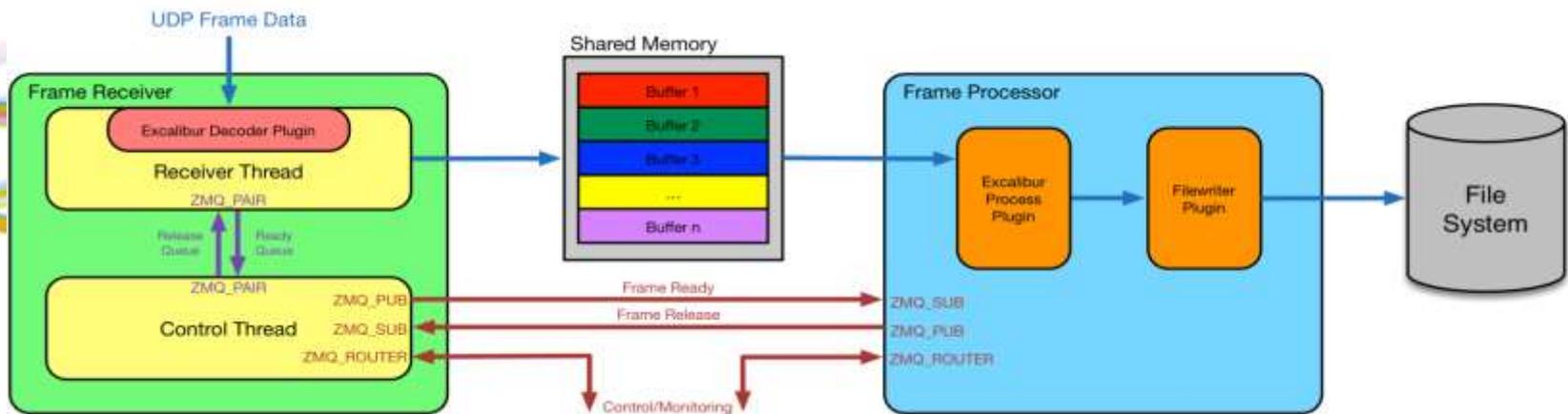

OdinData

- **FrameProcessor** receives message with frame pointer
- **Process plugin** performs relevant processing to produce an image
 - Descrambling detector specific data
- **Detector agnostic FileWriterPlugin** writes pre-chunked data directly to HDF5 dataset



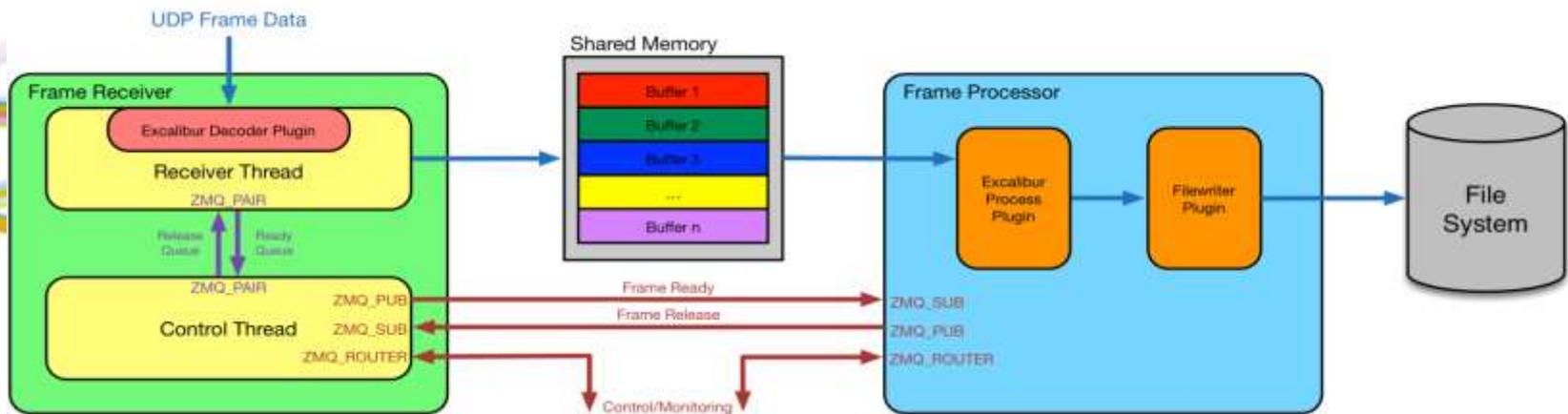
OdinData

- Write data using Direct Chunk Write
- Only the raw dataset for best performance
- A single, live, view of the data produced using Virtual Dataset and SWMR functionality



OdinData

- **Handle errors gracefully throughout**
 - Missing packets
 - Missing full frames
- **Random ordering**
 - Frames stored in correct order in HDF5



areaDetector Integration - ADOdin

- **Beamlines require areaDetector**
 - Integration through GDA and/or Malcolm for scanning
- **Utilize the public HTTP/REST API**
 - Just like any other control system from our collaborators
- **Same idea as ADEiger and ADPilatus**
 - Control and status monitoring
 - External servers/processes handle data stream

areaDetector Integration - ADOdin

OdinData Configuration

HDF5

File Directory: /tmp

Acq. ID: image_001

File Template: %s.hdf5

Frame Count: 10

Start [Green] Timeout Stop Advanced

Advanced HDF5 Configuration

Dataset Configuration

Image Width / Height: 2048 / 256

Chunk Depth / Width / Height: 1 / 2048 / 256

HDF5 Fill Value: 0.0

Data Type: UInt16

Compression Type: None

OdinData Status (One for each process)

OdinData Status

Data Frames: 3 / 10

Meta Frames Written: 0

Buffers Free: 0

Process Rank: 0

Full File Name: image_001_0010

Status: <ODINDETECTOR:OD1:Status>

Connected: [Green] Initialised: [Green] Writing: [Green] Queued: [Black]

File Configuration

Flush Period (Frames): 0

Initial Frame Offset: 0

Timeout Period: 0

Frames per Block: 1

Blocks per File: 0

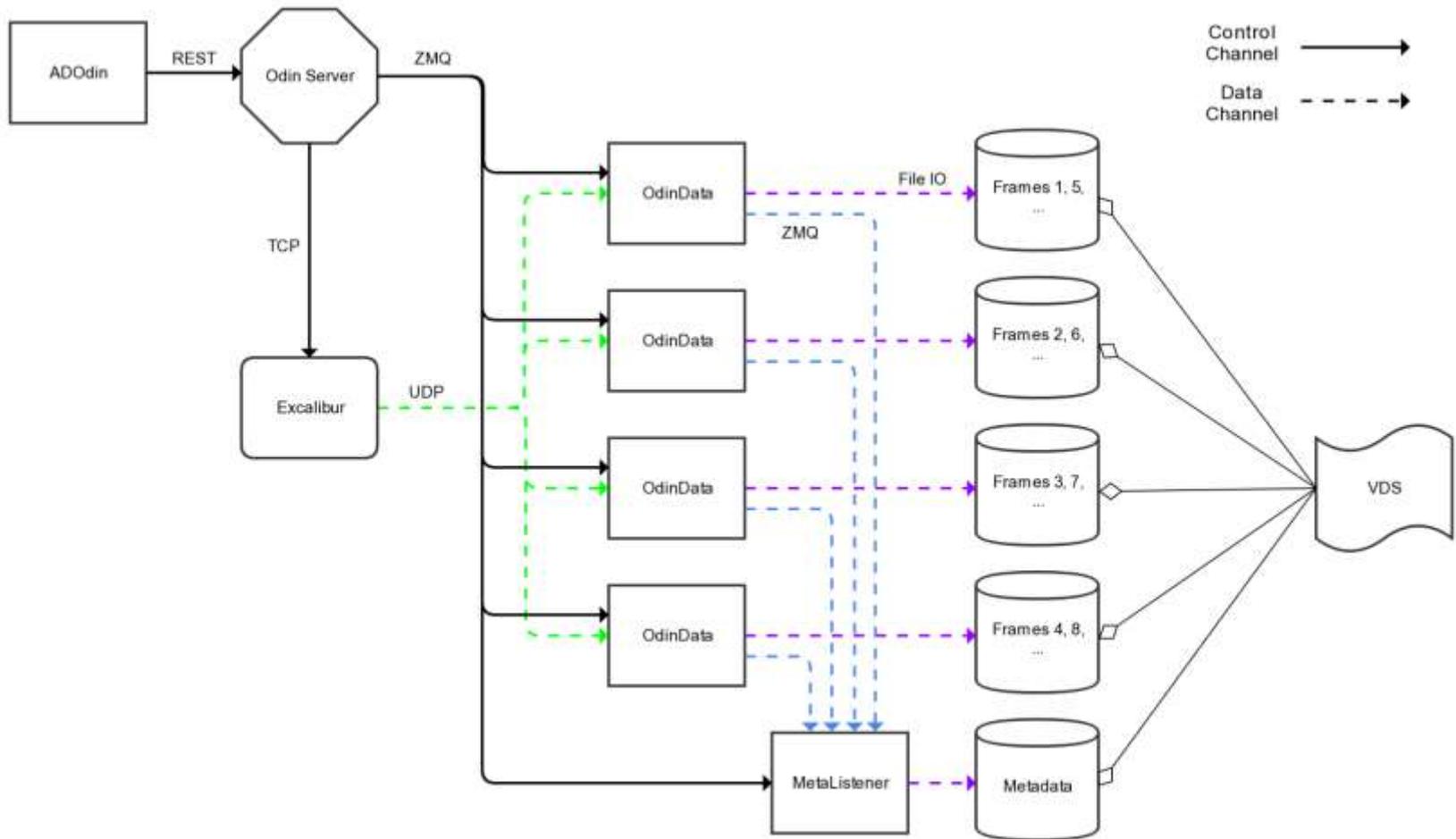
Chunk boundary alignment: 1 bytes

Chunk boundary threshold: 1 bytes

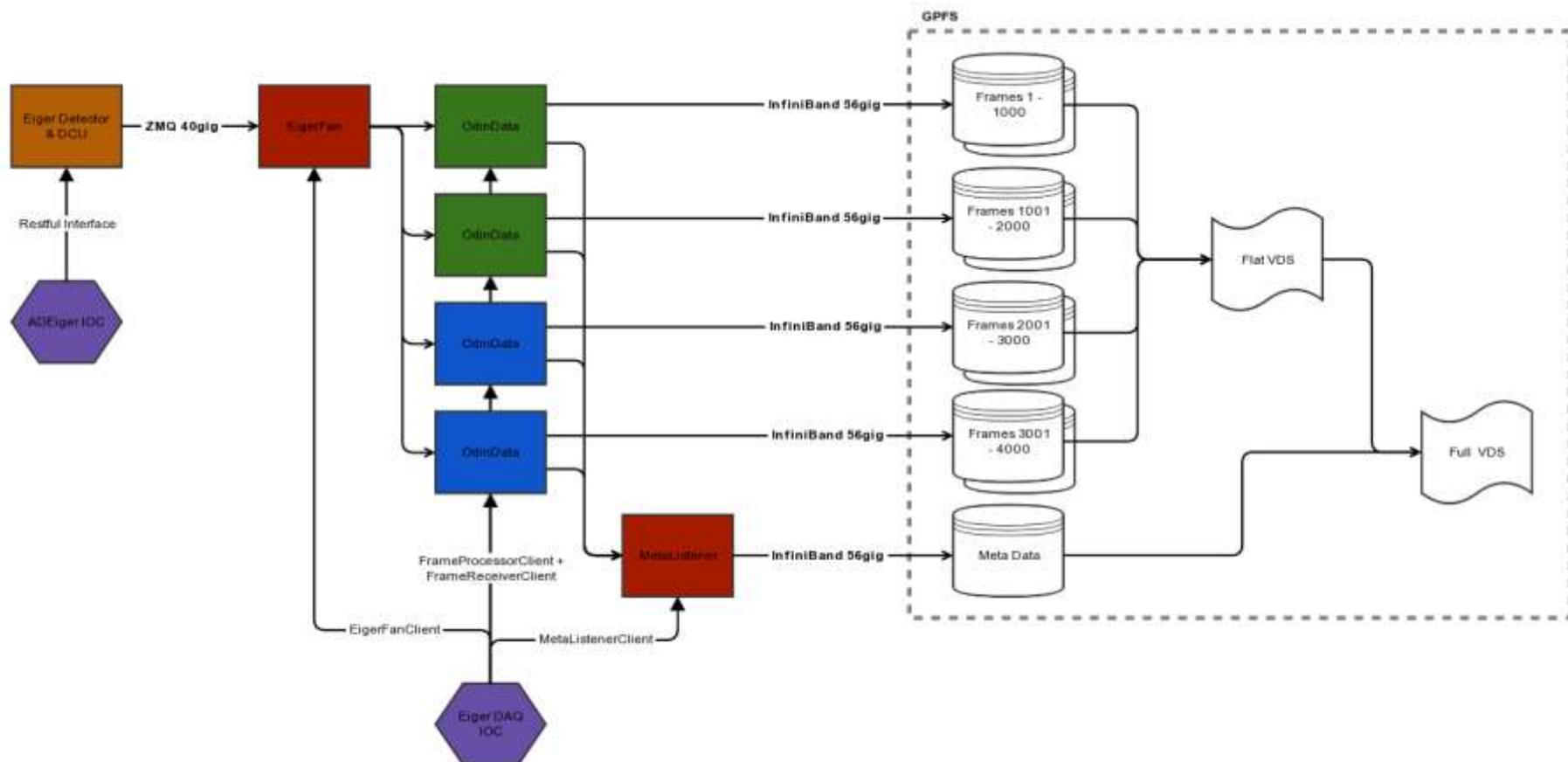
Earliest HDF5 Version: No



Excalibur



VMXi Eiger X 4M



iOZ-2-control

CS04R-9C-SERV-116

CS04R-9C-SERV-117

CS04R-9C-SERV-118

In Summary...

- **Odin is a control system agnostic single point of control**
- **Integrates with areaDetector for beamline application**
- **WIP but used on 2 detector systems**
 - Excalibur
 - Eiger 4M
- **Odin is our Detector Control & DAQ backend for in-house and collaborative detector systems**

Thanks

Questions?

Gary Yendell (Diamond)

Tim Nicholls (STFC)

Alan Greer (Observatory Sciences/Diamond)

Matthew Taylor (Diamond)

